

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ

Зеленский Святослав Дмитриевич

Выпускная квалификационная работа бакалавра

Распознавание автомобильных номеров

Направление 010400

Прикладная математика и информатика

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Сергеев С.Л.

Санкт-Петербург

2017

Содержание

Введение	3
Постановка задачи	5
Обзор литературы	6
Глава 1. Детектирование номера	7
1.1. Гистограммный анализ регионов	13
1.2. Метод Виолы-Джонса	15
1.3. Нейросетевой подход	17
Глава 2. Нормализация и сегментация	22
2.1. Нормализация	22
2.2. Сегментация	24
Глава 3. Распознавание символов	25
3.1. K-nearest	25
3.2. Использование нейросетей	26
Выводы	29
Заключение	30
Список литературы	31

Введение

Ежедневно миллионы людей используют автомобиль как средство постоянного передвижения, будь это такси или личный транспорт. Автомобильная промышленность активно развивается, и плодами их деятельности пользуется все больше и больше людей. Количество автовладельцев с каждым днем увеличивается. Статистика показывает, что каждый час на дороги выезжает новый водитель в новой машине.

В России ситуация обстоит точно также. Каждый день на дороги страны выезжает около 56 миллионов автомобилей. Для идентификации автомобиля его оборудуют двумя номерными знаками спереди и сзади, расположение и размеры которых регулируются ГОСТом.

Далеко не секрет, что количество дорожно-транспортных происшествий по причине нарушения правил дорожного движения растет пропорционально количеству автомобилей. Чаще всего, ДТП происходят по нескольким причинам: езда в нетрезвом виде, нарушение скоростного режима и выезды на полосу встречного движения. Если проблему езды в состоянии опьянения решить каким-то автоматическим комплексом затруднительно на данный момент (требуется полицейский, который остановит подозрительного водителя, оценит его состояние по объективным показателям), то за соблюдением рядности и скоростного режима уже сейчас по всему миру наблюдают автоматические комплексы.

Работают они по следующему принципу:

1. Фиксируют нарушение камерой
2. Распознают номер машины
3. Формируют постановление об административном правонарушении

После этого, инспектор просматривает материалы и принимает решение о привлечении автовладельца к ответственности.

Системы распознавания номеров не безупречны. Сейчас инцидентов с ошибочным определением номерных знаков стало гораздо меньше, однако

улучшать алгоритмы всегда есть куда.

Машинное зрение является одной из самых быстроразвивающихся технологий. Количество способов применения данной технологии сложно переоценить. Практически в любой сфере где есть какие-то графические данные можно использовать данную технологию. Параллельно с этим нейросети завоевывают все большие и большие высоты на различных мероприятиях, посвященных классификации объектов. Сверточные нейронные сети становятся трендом среди исследователей и разработчиков.

В целом, процесс распознавания автомобильных номеров состоит из трех этапов:

1. Детектирование номерного знака на изображение
2. Нормализация и сегментация номера на символы
3. Распознавание символов

В данной работе исследуются уже существующие решения на каждом из этапов и из композиции лучших предлагается новая реализация.

Постановка задачи

Цель работы: исследование процесса распознавания автомобильных номеров, оценка уже существующих методов и композиция лучшего решения с максимальной степенью точности.

План решения:

1. Рассмотреть алгоритмы выделения контура.
2. Проанализировать и выбрать оптимальный метод определения номера.
3. Улучшить изображение найденной номерной рамки до приемлемого качества.
4. Разбить полученный номер на символы.
5. Выбрать оптимальные методы для распознавания и посимвольно распознать номер.

Обзор литературы

В статье «Обзор методов фильтрации и сегментации цифровых изображений» Стругайло В.В. журнала «Наука и образование» МГТУ им.Баумана рассматриваются методы выделения контуров Собеля, Кэнни, Лапласиана гауссиана, Робертса и Прюетта. [1]

В издании Tamkang Journal of Science and Engineering, Vol. 13, No. 4, pp. 433-442 (2010) авторами Куо-Ming Hung и Ching-Tang Hsieh рассматривается метод нахождения номера с помощью гистограммного анализа регионов. [2]

Авторами Р. Viola и М. Jones в книге “Robust real-time face detection” 2004 рассматриваются алгоритмы детектирования лица с помощью примитивов Хаара. [3]

Про детектирование объектов с помощью нейросетей писали Hoiem, D., Chodpathumwan, Y., и Dai, Q. в 2012 году в журнале Diagnosing Error in Object Detectors. Computer Vision – ECCV, Springer Berlin Heidelberg, стр. 340–353. [4] Также данную тему поднимал Fatih Porikli и Tekin Kocak в 2007 году в своём исследовании «Robust License Plate Detection Using Covariance Descriptor in a Neural Network Framework». [7]

О нормализации и сегментации отлично написал в 2007 году Ondrej Martinsky в обзорной статье под названием «ALGORITHMIC AND MATHEMATICAL PRINCIPLES OF AUTOMATIC NUMBER PLATE RECOGNITION SYSTEMS». [5]

Алгоритм k-nearest досконально описан в книге издательства MIT 2005 года авторов Shakhnarovich Darrell и Indyk, eds. «Nearest-Neighbor Methods in Learning and Vision». [6]

Глава 1. Детектирование номера

Предобработка изображения

Изначально имеется изображение автомобиля, снятое под каким-то углом, часто загрязненное, иногда смазанное. Некоторые из рассматриваемых в дальнейшем методов требуют дополнительных преобразований входной картинке.

Стоит отметить, что все последующие методы работают с черно-белыми изображениями. То есть, сперва исходная картинка из RGB переводится в оттенки серого (рис.1).



рис. 1 Исходное изображение

Рассмотрим способы выделения контуров: операторы Собеля, Кэнни, Робертса и Превитта.

Оператор Собеля

Говоря простым языком, результатом работы данного дифференциального оператора в точке с постоянной яркостью является нулевой вектор, а если же она принадлежит границе областей с различной

яркостью, то вектор, пересекающий границу и направленный в сторону увеличения яркости. Формально его работу можно описать следующими формулами:

$$1. G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A$$

$$2. G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

$$3. G = \sqrt{G_x^2 + G_y^2}$$

, где A – исходное изображение.

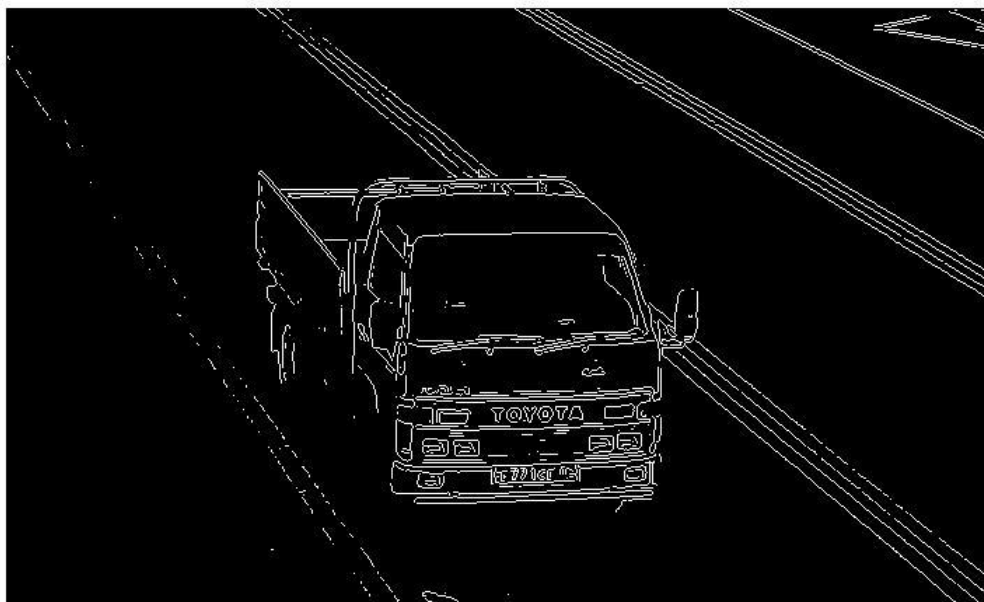


рис. 2 Оператор Собеля

Как видно на рисунке 2, оператор выделил границы. Стоит отметить, что на области автомобиля явно виден контур автомобильного номера и даже его символы.

Оператор Кэнни

В общем случае, работа данного оператора состоит из пяти этапов:

1. Сглаживание – изображение размывается для устранения шума.

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A$$

2. Поиск градиентов – контуры там, где перепад яркости максимален.

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$

θ округляется и принимает значения: 0, 45, 90, 135.

3. Подавление немаксимумов – только локальные максимумы определяются как границы.
4. Двойная пороговая фильтрация – здесь потенциальные границы задаются как пороги.
5. Трассировка области неоднозначности – все границы, не связанные с уже установленными границами, удаляются



рис. 3 Оператор Кэнни

Рассмотрев рисунок 3 видно, что данный фильтр для задачи нахождения номера избыточный – слишком много посторонних контуров он определяет. В дальнейшем, такое количество информации на изображении будет только вредить алгоритму.

Оператор Робертса

Этот оператор работает быстрее оператора Собеля:

$$G_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} * A$$

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$



рис. 4 Оператор Робертса

Для данного примера рамка номера и символы также ярко видны. Но если бы машина была грязная, то скорость вычисления сыграла бы не в пользу определения номерной рамки. Ведь по дорогам часто ездят запыленные автомобили с немытыми номерами. Для задачи детектирования номера данный оператор подходит только в случае высококачественного изображения с чистым номером.

Оператор Прюетта

В целом, он почти идентичен оператору Собеля, с тем лишь исключением, что у оператора Собеля центральные веса увеличены вдвое:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * A$$

$$G = \sqrt{G_x^2 + G_y^2}$$



рис. 5 Оператор Прюетта

Как видно на рисунке 5, границы номера также хорошо выделены, как и у оператора Собеля. Но взяв изображение более низкого качества, где лобовая поверхность машины снята под углом, не будет видно таких же ярких силуэтов у номерной рамки.

В среде разработки MATLAB все вышеуказанные методы уже реализованы.

```
>>I=imread('test.jpg');
```

```
>>imshow(I);
```

```
>>I=rgb2gray(I);
```

```
>>BW1=edge(I,'Roberts');
```

```
>> BW2=edge(I,'Sobel');
```

```
>> BW3=edge(I,'Prewitt');
```

```
>> BW4=edge(I,'Canny');

>> imshow(BW1)

>> figure, imshow(BW2)

>> figure, imshow(BW3)

>> figure, imshow(BW4)
```

1.1 Гистограммный анализ регионов

Первым и самым простым для понимания алгоритмом является гистограммный анализ регионов. Метод основан на том, что символы на номере, фон и рамка контрастны. Говоря простым языком, после выделения границ для изображения строится гистограмма. И номер находится там, где фиксируется максимум белых точек. Пример изображен на рисунке 6.

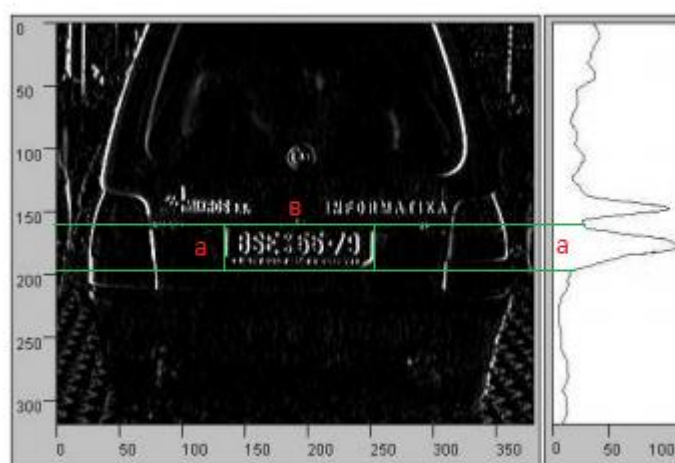


рис. 6 Гистограммный анализ регионов

Как видно, фиксируется глобальный максимум. Далее, зная пропорции номера, можно сделать вывод о длине рамки. После этого, производится поиск кандидата на номер – вертикальная линия, после которой следуют частые изменения яркости.

Размеры номера согласно ГОСТу составляют 520 миллиметров в длину и 112 миллиметров в ширину. Изображение из ГОСТа можно увидеть на рисунке 7:



рис. 7 ГОСТ номерного знака

Ширина относится к длине как 0.22.

На рисунке 6 расстояние a известно из проекции изображения, расстояние b можно высчитать: $b = \frac{a}{0.22}$.

От вертикальной границы номера переместившись на расстояние b . Это и будет искомой рамкой. После этого, полученный прямоугольник передается на следующий этап.

Очевидно, что данный алгоритм неустойчив к шумам и грязи. Кроме того, в случае большой перспективы (большого угла наклона камеры при фотосъемке) или маленьким изображением машины по сравнению с окружающими объектами, выделение максимума проекции будет затруднено и невозможно в принципе.

Подводя итог, данный алгоритм является простым в понимании и реализации, но работает только в случае фотографирования машины как главного объекта фотографии, без захвата посторонних предметов и под прямым углом. В любых иных случаях алгоритм не будет срабатывать. В реальных условиях данный алгоритм использоваться не может.

1.2. Метод Виолы-Джонса

Предыдущий метод никуда не годится в условиях реальных дорог. Грязь, снег, пыль – те спутники, без которых автомобильные номера редко встречаются.

В 2001 году Паул Виола и Майкл Джонс предложили алгоритм для нахождения лиц с использованием примитивов Хаара. Суть метода заключается в нахождении признаков на изображении, характерных для лица человека – глаза, нос, рот. Примитивы Хаара на рисунке 8 как раз и отражают в себе данные признаки.

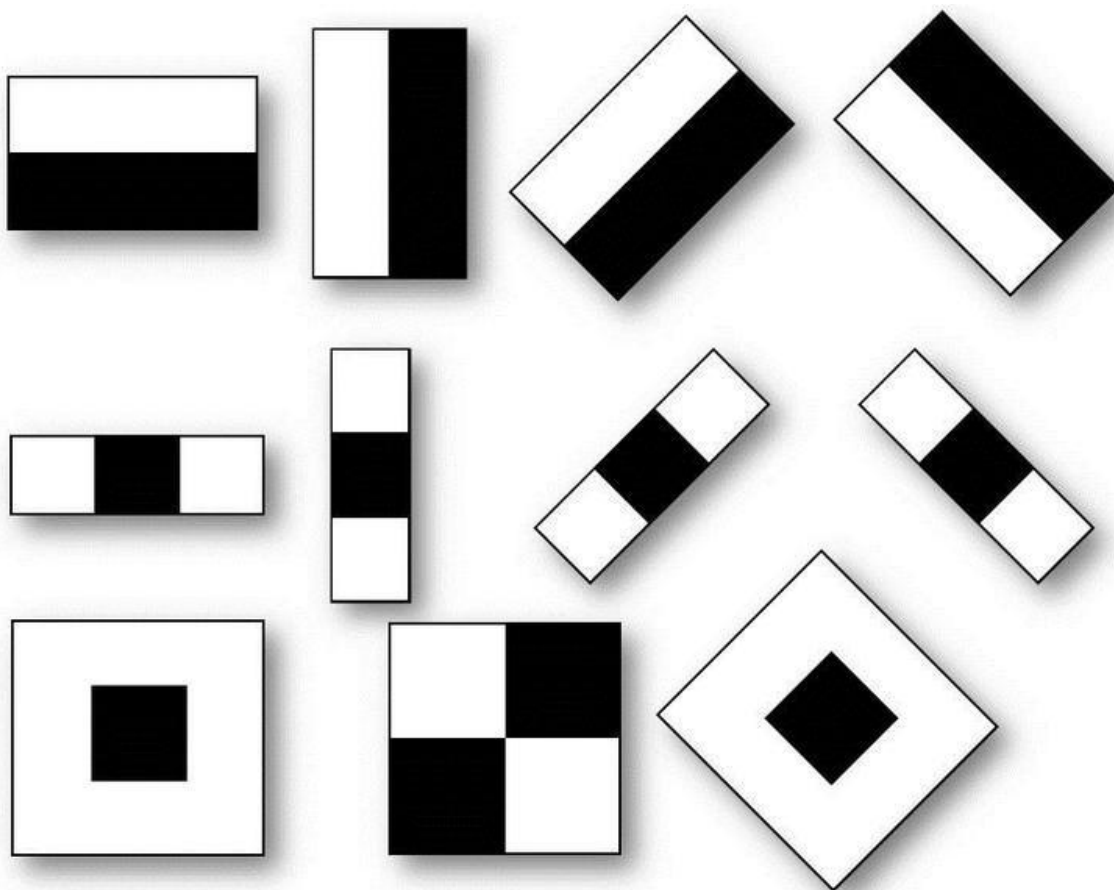


рис. 8 Примитивы Хаара

Попросту говоря, суть данного алгоритма в том, что по изображению двигается окно, к которому прикладываются данные примитивы поочередно.

В случае нахождения признаков, подходящих под условие задачи, данное окно фиксируется. Обученный каскад Хаара для задачи распознавания автомобильных номеров уже добавлен в библиотеку OpenCV – это библиотека алгоритмов обработки изображений, компьютерного зрения и численных алгоритмов общего назначения с открытым кодом.

Как алгоритм поиска номеров получались результаты около 90% верного определения. Работа алгоритма в принципе достаточно сложна. Основные этапы изображены на рисунке 9.

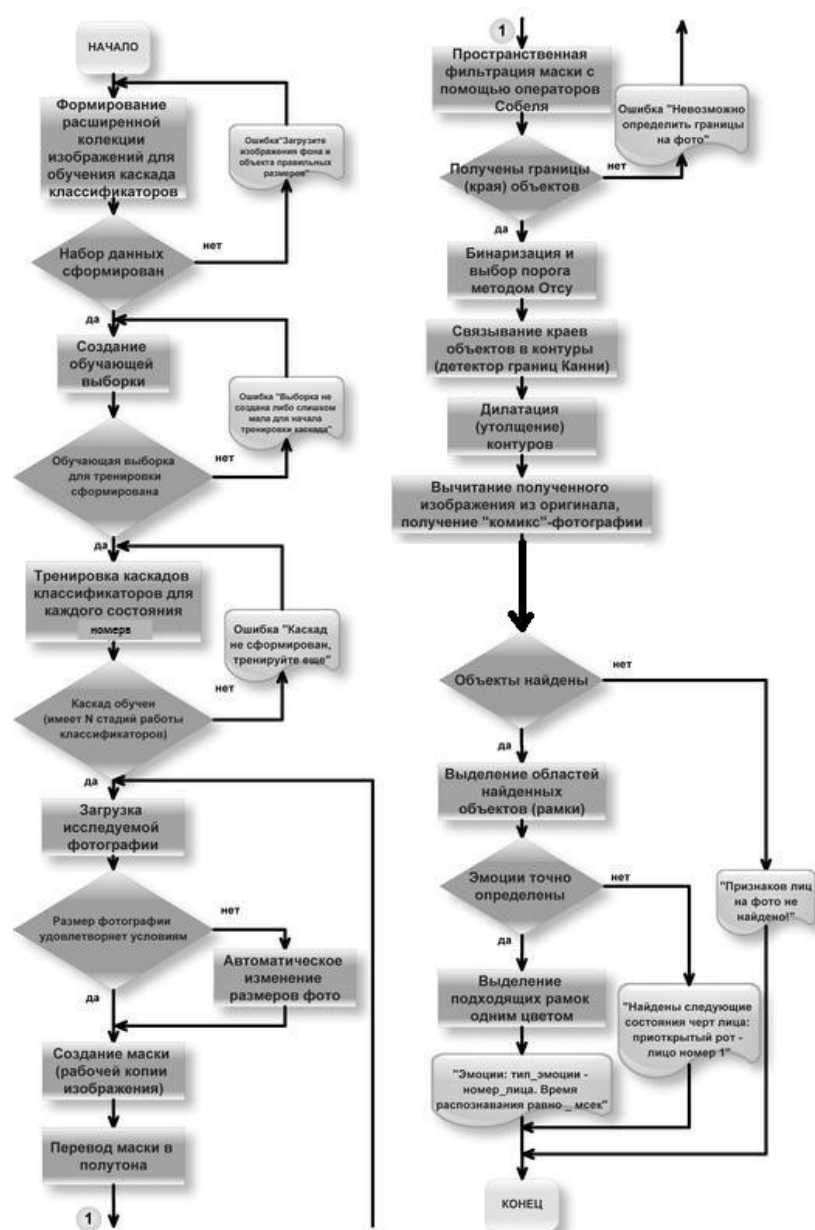


рис. 9 Обучение

1.3. Нейросетевой подход

Сегодня с помощью продукта компании Nvidia DIGITS исследователи могут с помощью нативного графического интерфейса решать проблемы подготовки данных, создания сверточной сети, наблюдения за процессом обучения прямо в момент работы приложения.

В версии DIGITS 4 реализуется новый подход к задаче нахождения объектов, который позволит обучить сети для идентификации различных тел, например, автомобильных номеров или автомобилей на аэрофотоснимках.



рис. 10 Результат работы сети DetectNet для обнаружения транспортных средств

На рисунке 10 показывается результат работы нейросети DetectNet, которую обучили обнаруживать транспортные средства на фотографиях аэрофотосъемки.

DetectNet в качестве обозначения класса использует трехмерный формат метки. Это позволяет работать алгоритму с изображениями различного размера и любым числом присутствующих объектов.

На рисунке 11 показан алгоритм обработчика изображений из обучающей выборки с разметкой для обучения сети DetectNet.

1. На изначальную картинку накладывается решетка с размером меньшим, чем самый малый объект, который подлежит обнаружению.
2. Каждый квадрат в решетке размечается классом объекта, находящимся внутри и координатами пикселя углов ограничивающего прямоугольника относительно центра квадрата решетки.
3. Если нет объектов в квадрате решетки, то квадрат помечается как “dontcare”. В формат данных для входа вносится значение “coverage” со значением 0 в случае отсутствия объекта и 1 в случае детектирования
4. Если в окно решетки попало больше одного объекта, то выбирается наибольший. Если размер пикселей одинаков, то выбирается с наименьшей координатой по оси Y.

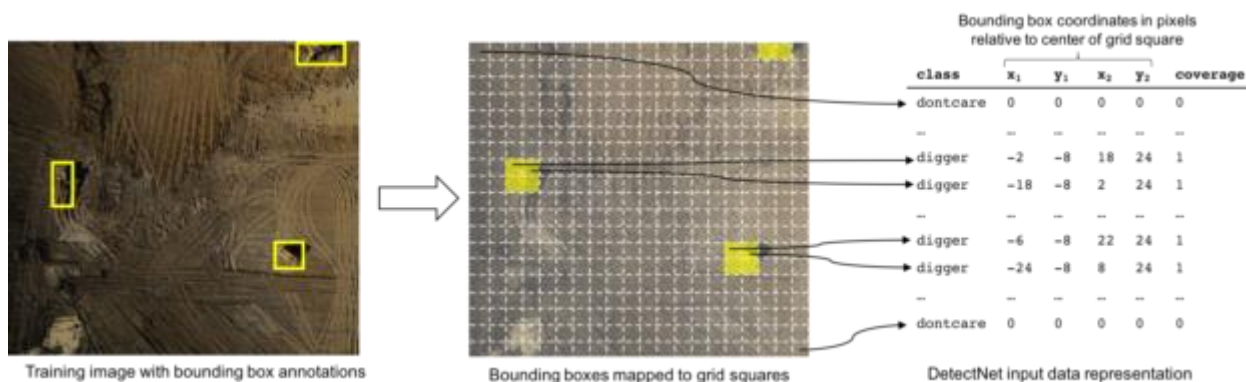


рис. 11 Представление входных данных для сети DetectNet

Говоря простыми словами, DetectNet на основе обучения старается предсказать есть ли объект в квадрате или нет.

Архитектура нейросети DetectNet для тренировки представлена на рисунке 12.

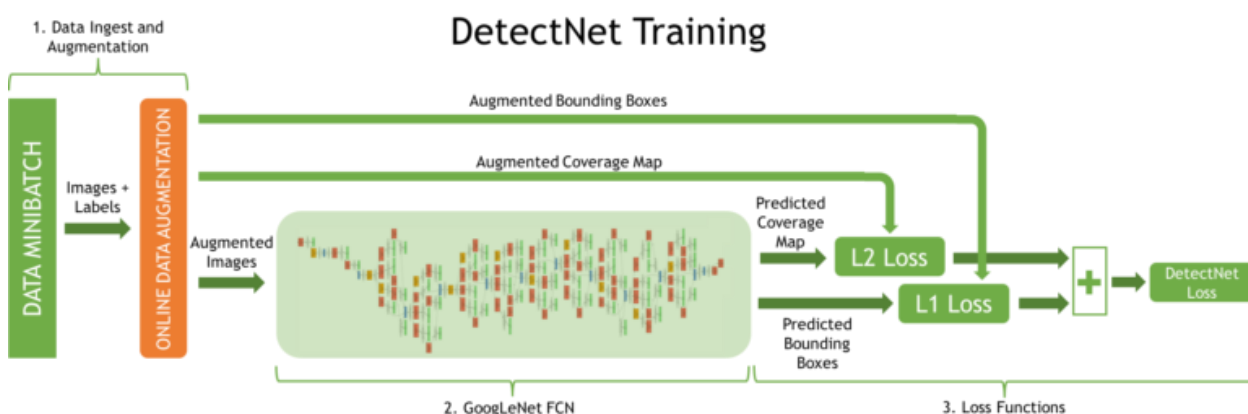


рис. 12 Структура сети DetectNet для обучения

Нейросеть основана на фреймворке Caffe и имеет 5 слоев.

1. На вход поступает картинка и метки
2. Полностью сверточная сеть (FCN) извлекает признаки, предсказывает классы объектов и размечает прямоугольники-ограничители.
3. Функции потерь, одновременно, считают ошибку по задачам предсказания покрытия объекта и углов ограничивающих прямоугольников по квадратам решетки.

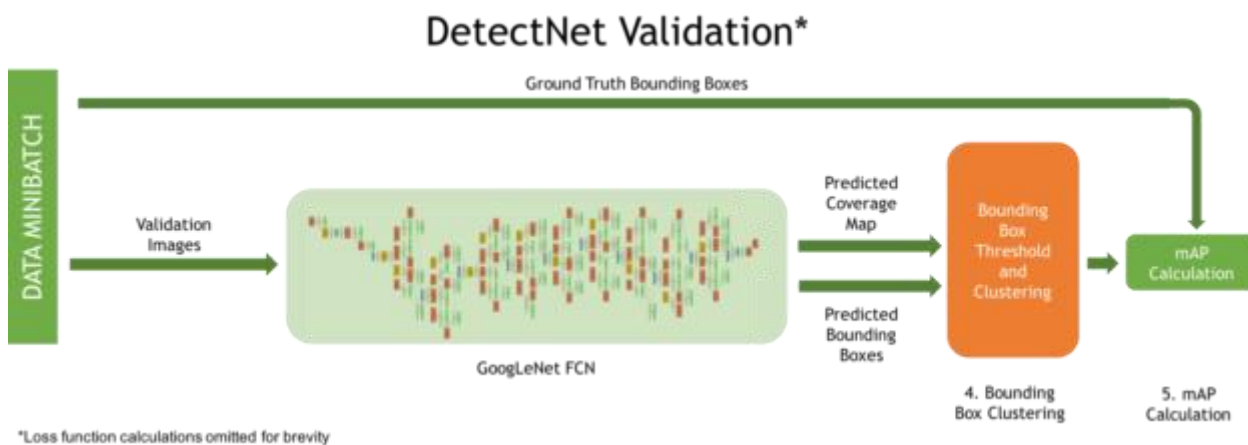


рис. 13 Структура сети DetectNet для проверки

4. Кластеризация ограничивающих прямоугольников, которые предсказала система
5. Подсчет упрощенной метрики mean Average Precision, характеризующая эффективность модели для тестовой выборки

Применяя данную технологию для нахождения номеров, можно получить весьма неплохой результат:

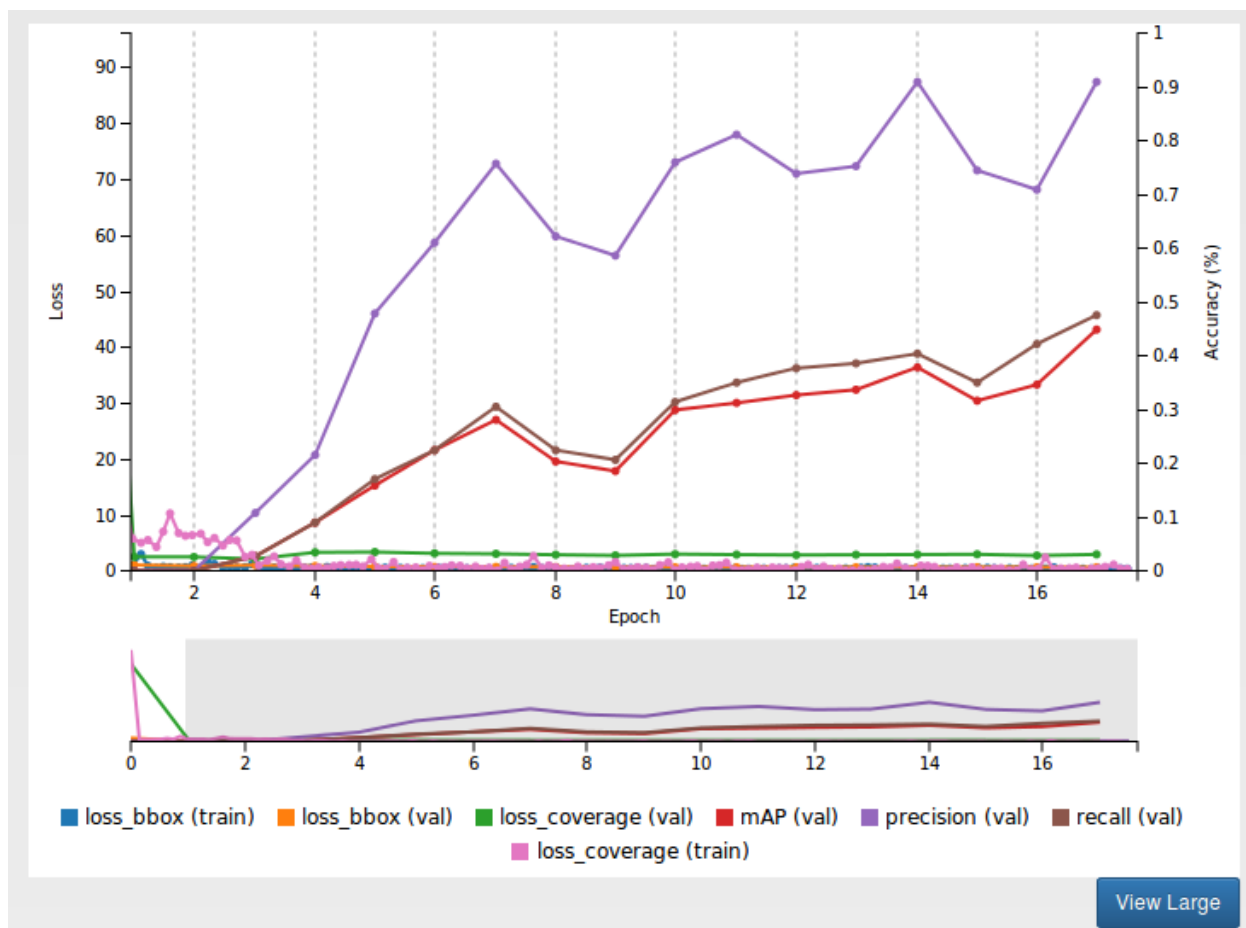


рис. 14 График работы системы

Как видно, уровень обнаружения достигает 91%.

Преимущества этого метода перед предыдущим является простота настройки, наличие графической оболочки, наличие демонстрационной модели «из коробки». Данный метод является уже готовым и запустить его получится в течение нескольких часов. Предыдущий же метод в случае не использования готовых библиотек, можно реализовывать на протяжении недель. Кроме того, Nvidia DIGITS не требует никаких дополнительных знаний в области computer science. Распространяется система бесплатно в

рамках научно-исследовательской деятельности. Разработчики постоянно работают над расширением функционала и совсем недавно выпустили обновленную систему DIGITS 5, в которой тренировка и распознавание происходит еще быстрее.

На выходе система выдает изображения с отмеченными на ней областями с номером:



рис. 15 Результат работы

Глава 2. Нормализация и сегментация

2.1. Нормализация

После предыдущего этапа на выход подается рамка изображения автомобильного номера, например, как на рисунке 16. Неизвестен градус наклона, возможна грязь на номере, изображение снято размыто – это лишь часть проблем, которые необходимо разрешить для дальнейшей работы системы.



рис. 16 Вывод программы после детектирования

Рассмотрим возможные пути решения проблем по порядку.

Для определения угла наклона изображения наиболее простой в реализации является преобразование Хафа. Этот метод находит прямые линии на изображении.



рис. 17 Преобразование Хафа

Далее выполняем поворот на найденный в предыдущем этапе угол, например, используя Matlab:

```
>>imrotate(Img,alpha);
```

Кроме этого, можно сразу по полученным после преобразования Хафа линиям обрезать изображение. Пример на рисунке 18.



рис. 18 Поворот и обрезание номерного знака

После выделения, необходимо сделать изображение более контрастным. Для этого можно линейно растянуть гистограмму изображения. В Matlab существует функция `imadjust(Img)`, которая производит данную операцию. Результат работы изображен на рисунке 19.



рис. 19 Увеличение контрастности номера

Теперь имеется правильно ориентированный черно-белый номер. Следующим этапом будет наложение фильтра LoG (Laplacian of Gaussian).

```
>> filt = fspecial('log',[7 7], 0.3);
```

```
>>srcImod = imfilter(Img,filt);
```

Оператор Лапласиана гауссиана задается по формуле:

$$LoG(x, y) = -\frac{1}{\pi * \sigma^4} * \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) * e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Результат на рисунке 20.



рис. 20 Применения оператора LoG

После данного этапа контуры букв четко видны и читаются на изображении. Далее требуется сегментировать номерной знак на символы.

2.2 Сегментация

После нормализации изображения сегментацию легче всего провести путем построения графика проекции на ОХ и выделения на графике максимумов. Пример работы можно увидеть на рисунке 21.

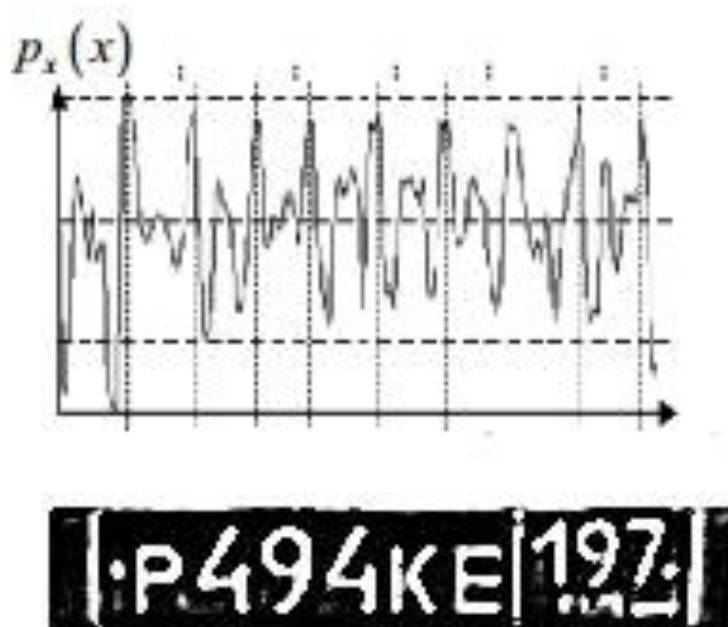


рис. 11 Построение проекции черных точек на ОХ

Естественно, у данного способа есть слабые места. Например, сильно зашумленный номер разделить на символы будет крайне проблематично, ведь приведенный метод сегментации зависит от цветов. Следовательно, при наличии черной грязи на белом номере кроме контура символов номера выделится и грязь.

Кроме того, существуют еще машины с изогнутыми номерами, а некоторые автовладельцы даже умудряются в квадратную полость американского номера «утопить» российский номерной знак.

Глава 3. Распознавание символов

Выходом прошлого этапа является 6 изображений цифр и 3 изображения букв. Данные картинки необходимо распознать. Благодаря стандартному шрифту использование тех или иных алгоритмов будет упрощено, ведь основные проблемы распознавания как раз у рукописных шрифтов.

Но ввиду того, что это все-таки цифры и буквы номерного знака автомобиля, часто информация может быть частично засорена шумом.

Существует множество алгоритмов с готовой реализацией для распознавания символов. Сравним два из них: k-nearest и нейросетевой.

3.1. k-nearest

Данный метод прост для понимания. Сперва необходимо набрать некоторое количество уже нарезанных букв, разделенных на классы. Часто создание такой базы является самой большой проблемой в виду колоссальных трудозатрат.

Далее задается мера расстояния между разбитыми классами. От выбранных значений будет зависеть точность последующего распознавания.

После этого рассчитывается расстояние от распознаваемого изображения до каждого класса поочередно. В случае равенства расстояния объект причисляется к более многочисленному классу.

Пример работы данного алгоритма можно рассмотреть на рисунке 22.

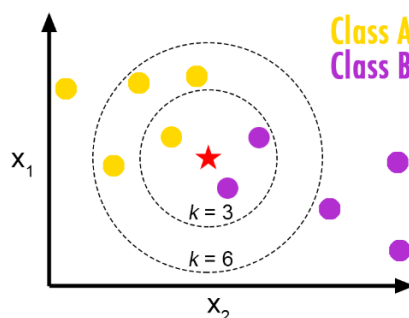


рис. 12 Работа k-nearest в задаче классификации

В случае $k=3$ объект причисляется в классу В, но в случае $k=6$ объект уже будет причислен к классу А.

Данный метод в случае набора значительного количества исходных данных можно использовать и в реальных условиях. На практике же, разметка и классификация необходимого количества символов с учетом всех особенностей весьма сложна. Теоретически, можно добиться приемлемых результатов, но база из уже размеченных символов должна быть больших размеров.

Плюсом является возможность отследить состояние на каждом шаге и вмешаться в распределение на классы в случае каких-то ошибок.

3.2. Использование нейросетей

Искусственная нейронная сеть – это математическая модель, а также её программная реализация, которая строится по принципу функционирования нервных систем живых объектов. С точки зрения машинного обучения, нейросеть представляет собой частный метод распознавания образов или классификации.

Простейшая модель нейронной сети изображена на рисунке 23.

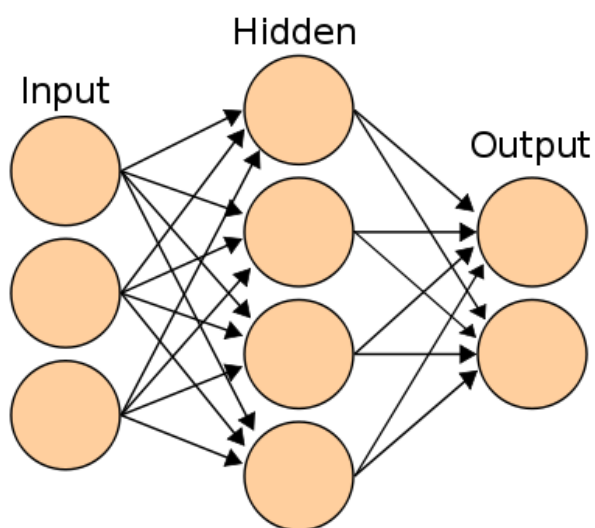


рис. 13 Пример простейшей нейросети

Нейросети обучаются – это их отличие от большинства других алгоритмов. Технически, обучение – это процесс нахождения коэффициентов связей между нейронами. Благодаря обобщению, нейросеть может сделать вывод об объекте, который не присутствовал в обучающей выборке. Кроме этого, проблема зашумленности также не является существенной. Нейронные сети идеально подходят для задачи распознавания автомобильных номеров.

В 2017 году построение нейросетей не является чем-то чрезвычайно сложным. Существует огромное количество библиотек, упрощающие те или иные аспекты создания. Например, для языка Python существует библиотека `keras`, с помощью которой за нескольких команд можно создать на устройстве сверточную нейронную сеть.

Вернемся к задаче распознавания символов на номерном знаке. Бинаризация изображения помогает сегментировать номер, но непредсказуемо искажает сами символы. Для построения нейросети необходимо учитывать этот факт.

В качестве эксперимента было создано и обучено две нейросети: для черно-белых изображений и для бинаризованных. По отдельности точность распознавания оказалась в пределах 88%.

Далее, был проведен эксперимент по комбинированию построенных нейронных сетей. Третья нейросеть показала результат лучший – 91%. Для её построения были соединены обе предыдущих нейронных сетей с половинчатыми весами. Данный результат уже достаточен для использования в реальных условиях.

Примерная архитектура нейросети для классификации объекта изображена на рисунке 24.

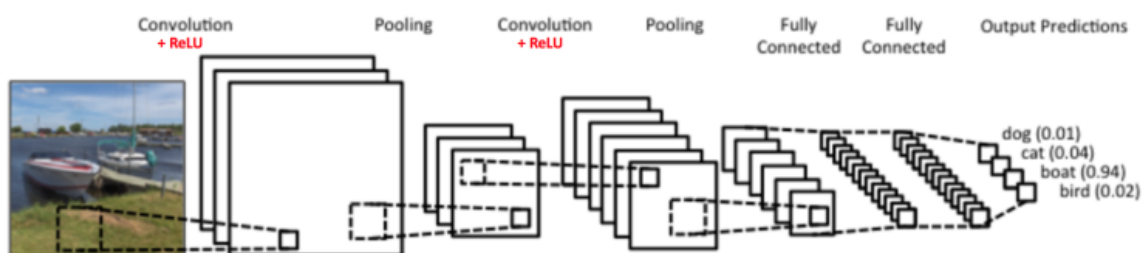


рис. 24 Архитектура сверточной нейросети

Данный метод также имеет свои минусы. Одним из таковых является невозможность отслеживания аномального поведения внутри сети.

Конечно, по сравнению с K-nearest, данный метод выигрывает. Нейросетевой подход хорош в случае реальных жизненных условий, но в качестве понимания процесса распознавания подойдет и K-nearest.

Выводы

В рамках проделанной работы был проведен анализ методов детектирования, нормализации и сегментации изображения, содержащего автомобиль с номерным знаком. Кроме этого, было проведено сравнение различных алгоритмов выделения контура для поиска номера – наиболее подходящем для этого является оператор Собеля. В качестве поисковых алгоритмов были рассмотрены методы гистограммного анализа регионов, Виолы-Джонса и Nvidia DIGITS. Лучшие результаты (91%) показал последний метод, который еще подкупает удобством и простотой настройки. В качестве распознающих методов были рассмотрены подробно два: K-nearest и сверточная нейросеть. У обоих методов свои минусы, но лучшим решением является использование комбинации из двух сверточных нейронных сетей, созданными на базе библиотеки keras. Результатом работы является определение с 91% точностью символов на номерном знаке.

Заключение

Задача распознавания автомобильных номеров имеет множество решений. На каждом этапе алгоритма существует большое количество путей, по которым может пойти разработчик. В данной работе сравнивались некоторые из них. С увеличением вычислительных мощностей область машинного зрения развивается быстрее и быстрее. Некоторые производители стали обращать внимания на потребности исследователей и производить действительно хорошие продукты, помогающие ученым.

Кроме того, теперь не надо прикладывать больших усилий для создания нейронных сетей. Практически на каждом языке программирования есть доступные библиотеки, позволяющие настроить и обучить нейросеть без погружения в глубину её реализации – все уже сделано сторонними разработчиками. Необходимо только прочесть документацию и подготовить необходимое обучающее множество.

Также были выявлены общая проблема для всех алгоритмов: грязные номера. Эту проблему можно решить только отчасти. Если какой-то из символов загрязнен целиком, то распознавание станет невозможным.

Список литературы

1. «Обзор методов фильтрации и сегментации цифровых изображений» Стругайло В.В., журнал «Наука и образование» МГТУ им.Баумана №5, май 2012.
2. Tamkang Journal of Science and Engineering, Vol. 13, No. 4, pp. 433-442 (2010) «A Real-Time Mobile Vehicle License Plate Detection and Recognition», Kuo-Ming Hung and Ching-Tang Hsieh
3. P. Viola and M. Jones. Robust real-time face detection. IJCV 57(2), 2004
4. Hoiem, D., Chodpathumwan, Y., and Dai, Q. 2012. Diagnosing Error in Object Detectors. Computer Vision – ECCV 2012, Springer Berlin Heidelberg, 340–353.
5. «ALGORITHMIC AND MATHEMATICAL PRINCIPLES OF AUTOMATIC NUMBER PLATE RECOGNITION SYSTEMS» ONDREJ MARTINSKY, 2007, Brno University
6. Shakhnarovich, Darrell, and Indyk, eds. (2005). Nearest-Neighbor Methods in Learning and Vision. MIT Press. ISBN 0-262-19547-X.
7. «Robust License Plate Detection Using Covariance Descriptor in a Neural Network Framework», Fatih Porikli, Tekin Kocak, 2007

Ссылка на документ в Интернете:

1. About OpenCV // OpenCV. URL: <http://opencv.org/about.html>
2. <https://habrahabr.ru/company/recognitor/blog/221891/>
3. <https://habrahabr.ru/post/310332/>
4. <https://devblogs.nvidia.com/parallelforall/detectnet-deep-neural-network-object-detection-digits/>
5. <https://devblogs.nvidia.com/parallelforall/digits-deep-learning-gpu-training-system/>
6. <https://habrahabr.ru/company/recognitor/blog/312472/>